

WHAT IS CLAIMED IS:

- 1 1. A computer implemented method comprising:
2 receiving requests for a shared resource, wherein
3 access to the shared resource is controlled with a
4 software lock;
5 detecting that the software lock is unavailable;
6 emitting an operating system trace hook in response to
7 the detecting, wherein the operating system trace hook
8 is adapted to record lock usage data; and
9 analyzing the recorded lock usage data.
- 1 2. The method of claim 1 wherein the lock usage data
2 includes a timestamp corresponding to each request, a
3 lock address corresponding to an address of the
4 software lock, a stack call chain for each process
5 making one of the requests at the time the
6 corresponding request was made, and a request count.
- 1 3. The method of claim 1 further comprising:
2 receiving a lock release request from a process that
3 currently holds the software lock;
4 determining whether other processes are currently
5 waiting for the software lock;
6 emitting a second operating system trace hook and
7 releasing the software lock in response to determining
8 that other processes are currently waiting for the
9 software lock, wherein the second operating system

10 trace hook is also adapted to record lock usage data;
11 and
12 releasing the software lock without emitting a trace
13 hook in response to determining that other processes
14 are not waiting for the software lock.

1 4. The method of claim 1 further comprising:
2 identifying one or more processes causing lock
3 contention in response to the analysis of the lock
4 usage data.

1 5. The method of claim 4 further comprising:
2 modifying the identified processes' usage of the
3 shared resource.

1 6. The method of claim 1 further comprising:
2 detecting that the software lock is available in
3 response to receiving the request;
4 incrementing a lock counter that tracks the number of
5 times the shared resource was requested; and
6 setting ownership of the software lock to an
7 identifier corresponding to an identifier of the
8 requesting process.

1 7. The method of claim 6 wherein the setting ownership
2 further includes:
3 requesting that a lock ownership field of the software
4 lock be set to the requesting process identifier; and
5 receiving a completion code indicating that the
6 ownership has been set, wherein the incrementing of

7 the lock counter is performed in parallel with the
8 setting ownership by incrementing the lock counter
9 after the request for the lock ownership field has
10 been set and before the completion code has been
11 received.

1 8. An information handling system comprising:
2 one or more processors;
3 a memory accessible by the processors;
4 one or more shared resources accessible by the
5 processors;
6 a nonvolatile storage device accessible by the
7 processors;
8 one or more software locks, stored in the memory, for
9 controlling access to the shared resources;
10 an operating system that controls a plurality of
11 processes performed by the processors, the operating
12 system including a trace hook facility; and
13 a software lock measurement tool for measuring lock
14 contention, the software lock measurement tool
15 including:
16 means for receiving requests from one or more of
17 the processes for one of the shared resources,
18 wherein access to the shared resource is
19 controlled with one of the software locks;
20 means for detecting that the software lock is
21 unavailable;
22 means for emitting an operating system trace hook
23 in response to the detecting, wherein the

24 operating system trace hook is adapted to record
25 lock usage data; and
26 means for analyzing the recorded lock usage data.

1 9. The information handling system of claim 8 wherein the
2 lock usage data includes a timestamp corresponding to
3 each request, a lock address corresponding to an
4 address of the software lock, a stack call chain for
5 each process making one of the requests at the time
6 the corresponding request was made, and a request
7 count.

1 10. The information handling system of claim 8 further
2 comprising:
3 means for receiving a lock release request from a
4 process that currently holds the software lock;
5 means for determining whether other processes are
6 currently waiting for the software lock;
7 means for emitting a second operating system trace
8 hook and releasing the software lock in response to
9 determining that other processes are currently waiting
10 for the software lock, wherein the second operating
11 system trace hook is also adapted to record lock usage
12 data; and
13 means for releasing the software lock without emitting
14 a trace hook in response to determining that other
15 processes are not waiting for the software lock.

1 11. The information handling system of claim 8 further
2 comprising:

3 means for identifying one or more processes causing
4 lock contention in response to the analysis of the
5 lock usage data.

1 12. The information handling system of claim 8 further
2 comprising:

3 means for detecting that the software lock is
4 available in response to receiving the request;

5 means for incrementing a lock counter that tracks the
6 number of times the shared resource was requested; and

7 means for setting ownership of the software lock to an
8 identifier corresponding to an identifier of the
9 requesting process.

1 13. The information handling system of claim 12 wherein
2 the means for setting ownership further includes:

3 means for requesting that a lock ownership field of
4 the software lock be set to the requesting process
5 identifier; and

6 means for receiving a completion code indicating that
7 the ownership has been set, wherein the means for
8 incrementing the lock counter is performed in parallel
9 with the means for setting ownership by incrementing
10 the lock counter after the request for the lock
11 ownership field has been set and before the completion
12 code has been received.

1 14. A computer program product stored on a computer
2 operable media for measuring lock contention, said
3 computer program product comprising:

4 means for receiving requests for a shared resource,
5 wherein access to the shared resource is controlled
6 with a software lock;

7 means for detecting that the software lock is
8 unavailable;

9 means for emitting an operating system trace hook in
10 response to the detecting, wherein the operating
11 system trace hook is adapted to record lock usage
12 data; and

13 means for analyzing the recorded lock usage data.

1 15. The computer program product of claim 14 wherein the
2 lock usage data includes a timestamp corresponding to
3 each request, a lock address corresponding to an
4 address of the software lock, a stack call chain for
5 each process making one of the requests at the time
6 the corresponding request was made, and a request
7 count.

1 16. The computer program product of claim 14 further
2 comprising:

3 means for receiving a lock release request from a
4 process that currently holds the software lock;

5 means for determining whether other processes are
6 currently waiting for the software lock;

7 means for emitting a second operating system trace
8 hook and releasing the software lock in response to
9 determining that other processes are currently waiting
10 for the software lock, wherein the second operating

11 system trace hook is also adapted to record lock usage
12 data; and
13 means for releasing the software lock without emitting
14 a trace hook in response to determining that other
15 processes are not waiting for the software lock.

1 17. The computer program product of claim 14 further
2 comprising:

3 means for identifying one or more processes causing
4 lock contention in response to the analysis of the
5 lock usage data.

1 18. The computer program product of claim 17 further
2 comprising:

3 means for modifying the identified processes' usage of
4 the shared resource.

1 19. The computer program product of claim 14 further
2 comprising:

3 means for detecting that the software lock is
4 available in response to receiving the request;
5 means for incrementing a lock counter that tracks the
6 number of times the shared resource was requested; and
7 means for setting ownership of the software lock to an
8 identifier corresponding to an identifier of the
9 requesting process.

1 20. The computer program product of claim 6 wherein the
2 means for setting ownership further includes:

3 means for requesting that a lock ownership field of
4 the software lock be set to the requesting process
5 identifier; and
6 means for receiving a completion code indicating that
7 the ownership has been set, wherein the means for
8 incrementing the lock counter is performed in parallel
9 with the means for setting ownership by incrementing
10 the lock counter after the request for the lock
11 ownership field has been set and before the completion
12 code has been received.